A Design and Solving LPP Method for Binary Linear Programming Problem Using DNA Approach

S.Mohanambal, G.Sudha, A.Pethalakshmi, and R.Rajarajeswari

Department of Computer Science, M.V.M.Govt Arts college, Dindigul.
Email: mohanapranav@yahoo.com
Email: g.sudha79@rediff.com
Email: pethalakshmi@yahoo.com
Email: ganesh_akash@yahoo.com

Abstract

Molecular computing is a discipline that aims at harnessing individual molecules for computational purposes. This paper presents the applied Mathematical sciences using DNA molecules. The Major achievements are outlined the potential advances and the challenges for the practitioners in the foreseeable future. The Binary Optimization in Linear Programming is an intensive research area in the field of DNA Computing. This paper presents a research on design and implementation method to solve an Binary Linear Programming Problem using DNA computing. The DNA sequences of length directly represent all possible combinations in different boxes. An Hybridization is performed to form double strand molecules according to its length to visualize the optimal solution based on fluorescent material . Here Maximization Problem is converted into DNA computable form and a complementary are found to solve the problem and the optimal solution is suggested as per the constraints stipulated by the problem.

Keywords:

DNA Computing, Hybridization, Olingonucleotides

I. Introduction

An Integer Programming Problem in which all variables are required to be integer is called a Pure Integer Programming Problem. If some variables are restricted to be integers and some are not then the problem is a mixed Integer Programming Problem. The case where the Integer variables are restricted to be 0 or 1. Such problems are called Pure(Mixed) Binary Programming Problems or Pure(Mixed) Integer Programming Problems. The Binary Integer Programming Problem is a special form of an Integer Programming Problem in which the value of variable x_i is only 0 or 1. In this condition x_i can be referred to as either a “ Binary” or “ 0 – 1” variable.

Its general form can be defined as :

DOI : 10.5121/ijist.2012.2202

Max(Min) $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

Subject to the constraints

$a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n <=(>=) \; b_1$

$a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n <=(>=) \; b_2$

…. …. …. …. …. ….

$a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n <=(>=) \; b_m$

where , $x_i = 0$ or 1, $1<= i <= n$ and b_j are non- negative integers, $1<= j <= m$.

This model to solve the general Binary Programming Problem with DNA, when a_{ij} is an integer, each constraint subjected to optimization function can be transformed into corresponding constraints where a_{ij} takes the value as 0 and 1[13].

II. Literature Survey

This section discusses the Review of Literature related to DNA.

Leonard M.Adleman, Described DNA computing is a new computational paradigm that uses DNA macro molecules to solve computational problems. The main advantage of DNA over traditional electronic computers is that it is tiny, cheap and can react faster than silicon. Using standard processes and enzymes from biochemistry, a basic set of operations on DNA like cutting, ligation, separation and amplification can be implemented. These basic operations can be used to program a DNA based computer. This report presents a instance of the directed Hamiltonian path problem was solved using DNA macromolecules[1]. It was the first time a computation was performed on a molecular level. This was the beginning of DNA computing.

Yashida, H.Suyama, proposed the extraordinary computation parallelism, energy efficiency and information density inherent in molecular computing has encouraged the expectation that a DNA computer might be a type of next- generation computer [29].

Adleman and R.J.Lipton adopted a brute-force search strategy to solve NP-complete problems by DNA computing. A DNA data pool containing the full solution space must first constructed in the initial test tube(t_o) and then correct answers are extracted and/or false ones are eliminated from the data pool step by step. Thus, the number of distinct DNA strands contained in the initial test tube(t_o) grows exponentially with the size of the problem. The number of DNA strands required for large problems eventually swamps the DNA data storage, which makes molecular computation impractical from the outset. Lipton's Brute-force search DNA algorithm is limited about 60 to 70 variables and thus it is believed that DNA computers that use a brute-force search algorithm can not exceed the performance of electronic computers. Since then, studies on DNA computing have focused on reducing the size of the data pool [22].

E.Anne,W.Cai,and M.Robert et al. explained the field of DNA computing is concerned with possibility of performing computational using biological molecules. It provides an understanding how complex biological molecules process information in an attempt to gain insight into new models of computing. DNA computer is interested in applying computer science methods and models to understand such biological phenomena and gain interest into early molecular evolution and origin of biological information processing. In addition to classical method meant for solving integer programming problem, some molecular computing models are discussed and are based on primary trends in research studies known as solution based and surface based DNA computations[4].

J.E.Hopcroft et al. described in implementing for finite state machines with DNA computing, this model, the size of the molecules representing the finite state control depends on the length of the input string. In addition, the only limitation on sequences of input strands corresponding to the alphabet and the state of finite machine are related to error minimalisation considerations[18].

H.Y.Wu, proposed computers have obvious limits in storage, speed, intelligence and miniaturization. The methods of DNA computation have arisen, especially for their efficient parallelism. In order to solve the practical issue, there are still some problems that need further study in biologic technology. In this article, we highlight a DNA computing model to solve a problem of 0-1 programming problem. This model we proposed has a potential to solve the linear programming problem, which is an important issue in operations research [30] .In this method we adopt the fluorescence marking technique and laser focus technique and determine the solution by analyzing fluorescence, the method of which has some significant advantages such as low cost, low error, short operating time, reusable surface and simple experimental.

S.L.Gass described DNA computing is a novel method of solving a class of intractable computational problems. In which the computing speeds up exponentially with the problem size. Up to now, many accomplishments have been made to improve its performance and increase its reliability. In this paper we solved the general form of linear programming problem with fluorescence labeling techniques based on surface chemistry by attempting to apply DNA computing to a programming problem. Our method has some significant advantages such as simple encoding, low cost and short operating time [13].

M.Garzon and E.Eberbach take a different course of action to understanding the power of DNA computing by examining its relationship to low level complexity classes. In particular, explore the recognition of regular languages, a well known and understood complexity class with a wide variety of very practical applications. The design in this paper are intended to serve as a generic algorithm for implementation of a deterministic finite state machine using DNA processes[12]. In DNA based computation, the instances of a problem are encoded in oligo nucleotides or strands of DNA. The oligo nucleotides bind in an anti parallel way with respect to the chemically distinct ends, 5` and 3`, of the DNA molecule and also explore the fundamental processing capabilities of DNA computing.

L.Streyer et al. described the set of transition molecules defining the library, we consider what structures may be built with them alone. Given a finite state machine of interest , we create a massive number of its transition molecules through PCR[27]. Combining the reaction conditions favorable for hybridization of length and results in concatenations of transition molecules having WC – complementary target state encoding and source state coding strands. However this set of molecules represents a set of computation paths on the finite state machine of interest, none of these paths stem from an input string .

Eliezer L.Lozinskii presented the basic idea was that we construct a set of Boolean functions representing routing constraints and invoke a quantum Boolean SAT solve on the generated function to find any satisfying assignments. The found SAT solution determines a full detailed routing solution. DNA satisfiability based on DNA properties. The Boolean satisfiability problem is a decision problem considered in the complexity theory. DNA search algorithms used to find the required routing solutions more quickly and effectively [11].

Gi-Joon Nam et al. proposed a faster approach for finding the FPGA routing solution using DNA computing. Because the DNA computing, due to its high degree of parallelism can overcome the difficulties that may cause the problem intractable on silicon computers. However using DNA computing principles for solving simple problems may not be suggestible To make the DNA computing applicable in practice further research in both fields Computer science and Biology is necessary. Computer science needs to develop more elaborate DNA algorithms, which better enzymes and protocols are needed to from biology to manipulate DNA molecules more selectively with minimal errors[15].

R.Deaton et al. described the implementation of evolutionary algorithms in bio molecules would bring full circle of the biological analogy and present an attractive alternative to meet large demands for computational power[10].This paper, a review of the most important advances in bio molecular computing in the last few years were presented.

III. Model Representation

This model involves a system of equations that contains n variables x1,x2,………,xn and m equations. Each variable is represented by a Single stranded DNA stretch with a Double stranded tag at the beginning. This imparts a sticky end to each variable as shown in the following Figure.

X_i = Box ATTGCTAT
X_i^1 = Box TAACGATA

Representation of variables using DNA strands

a) Operating Principle

Assume the structures with different composition of nucleotides and tags are to be taken which denote the false values, $x_1,x_2,……..,x_n$. The constraints are provide to the solution space by using $x_1^{11},x_2^{11},…….,x_n^{11}$ strands which are complementary to the Single stranded portion of the variables $x_1,x_2,…….,x_n$. There $x_1^{11},x_2^{11},…….,x_n^{11}$ attach to their respective complementary portions on the variables which are $x_1,x_2,……x_n$. With the help of fluorescent tagged material I can readout our required solution. By using $x_1^{11},x_2^{11},………,x_n^{11}$ we provide all the given constraints in different pools where each pool satisfy one of the given constraints and screen out our solution space to a list of feasible solutions. Then every value of objective function is compared to every feasible solution to get an optimal solution.

IV. ALGORITHM

The following Algorithm used to solve the Binary Programming Problem.

STEP 1:

Generate all possible combinations of variable 0 or 1 in the given problem.

STEP 1a:

Combine a set of Single – Stranded DNA molecules that represent all variables in the computational problem at hand. Synthesize and place samples in an addressed fashion on a surface and arrange these Single – Stranded DNA molecules according to the form of dot matrix. DNA Oligonucleotides are tagged with two different fluorescent colors as DNA probes.

STEP 2:

Reject infeasible solutions according to constraint inequalities (reserved feasible solution).

STEP 2a:

For each inequality, by adding the corresponding complementary strand to the surface, solution that satisfies this inequality will be hybridized by a complementary strand that is tagged with a fluorescent label, with a differential value (D-value) of two different colors that is at least (not exceeding) bi. Further, we can determine the solution for satisfying (dissatisfying) constraint conditions by a method of fluorescence imaging.

STEP 3:

Generate remaining solutions.

STEP 3a:

The temperature is raised to separate all Double – stranded DNA into Single – strands by thermal denaturizing. The surface is returned to the initial state by washing in a buffer that is made of 10 μM Tris- Hcl, 5 μM Kcl, 5 μM Mgcl2, 10 μM SDS and 50 μM H_2o (without regard for infeasible solution determined in STEP 2a).

STEP 4:

Repeat steps 2 and 3. We can remove all infeasible solutions and obtain feasible solutions of the problem; then proceed step5.

STEP 4a:

Repeating steps 2a and 3a, We can reject all infeasible solutions and obtain a feasible solution of the problem then proceed step 5a.

STEP 5:

By comparing to value of object function corresponding to every feasible solution, we can obtain an optimum solution.

STEP 5a:

By calculating and comparing the value of the object function corresponding to every feasible solution, an optimum solution can be obtained.

V.HYPOTHETICAL ANALYSIS

Consider the following problem

$$\begin{aligned} &\text{Max} \quad U = 4x1 + 5x2 + 4x3 \\ &\text{Subject to the constraints} \\ &x1 - x2 - x3 <= 2 \\ &x1 + x2 + x3 <= 1 \\ &x1 + x2 + x3 <= 3 \\ \text{where } &x1,x2,x3 = 0,1.(x1= x, x2= y, x3= z) \end{aligned}$$

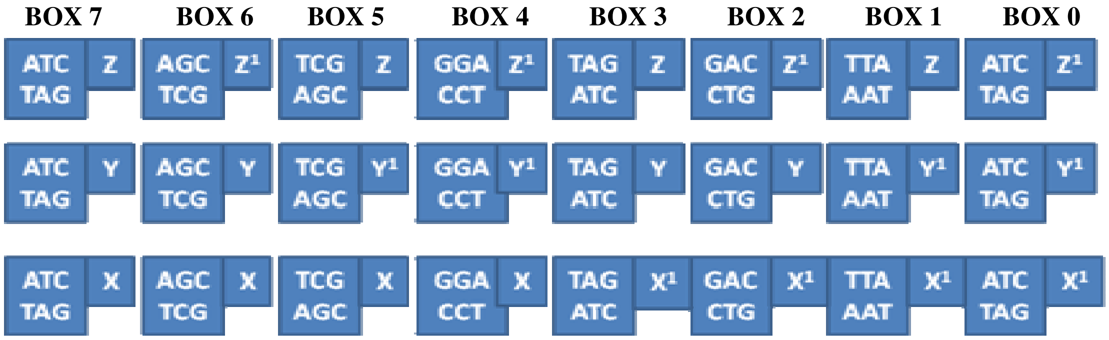
The process for solving Binary Programming Problem is divided into the following six steps.

STEP 1:

In a container we first synthesized 9 oligonucleotides, which were divided into 3 groups. The oligonucleotides of the first group represented variables x1,x2,x3 attached with different Boxes, The oligonucleotides of the second group similarly represented variables $x1^1,x2^1,x3^1$; also attached with a different Boxes (x1 = 1 if and only if $x1^1 = 0$, such as x2,x3); The oligonucleotides of the third group represented the complementary strands of the first group (without any Boxes) and are denoted as $x1^{11}$, $x2^{11}$, $x3^{11}$.

STEP 2:

We generate different combination of DNA molecules where we choose oligonucleotides x1,x2,x3 and $x1^1,x2^1,x3^1$ such that they must be very different, oligonucleotide x represent variable x1 = 1 and oligonucleotide $x1^1 = 0$, for x2,x3.



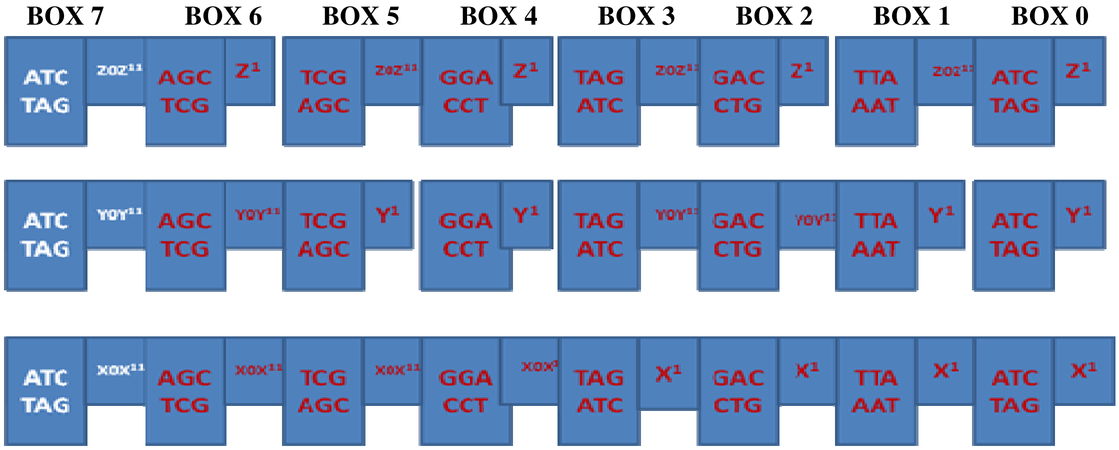
Combination of oligonucleotides are placed in a container

STEP 3:

Copies of the first container molecules are placed in 3 different containers for 3 equations. Total process is done in parallel and take less time.

STEP 4:

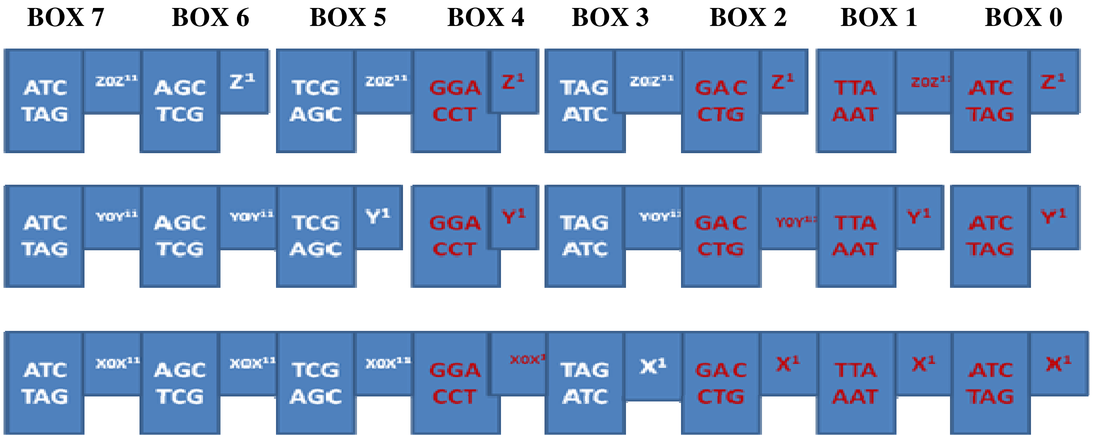
According to first equation, we added DNA probes, respectively.Tagging 3 oligonucleotides $x1^{11}$, $x2^{11}$, $x3^{11}$ with fluorescent material (chemical compound and green in color). For first constraint equation, we passed the complementary strands $x1^{11}$,$x2^{11}$,$x3^{11}$ tagged with fluorescent material corresponding to variable x1,x2,x3. Any solution satisfying this inequality in hybridized with at most 2 complementary strands tagged with fluorescent material (at least 2 two bright point) and the feasible solution of the problem is “6,5,4,3,2,1,0”.



Hybridized of the First constraint equation

STEP 5:

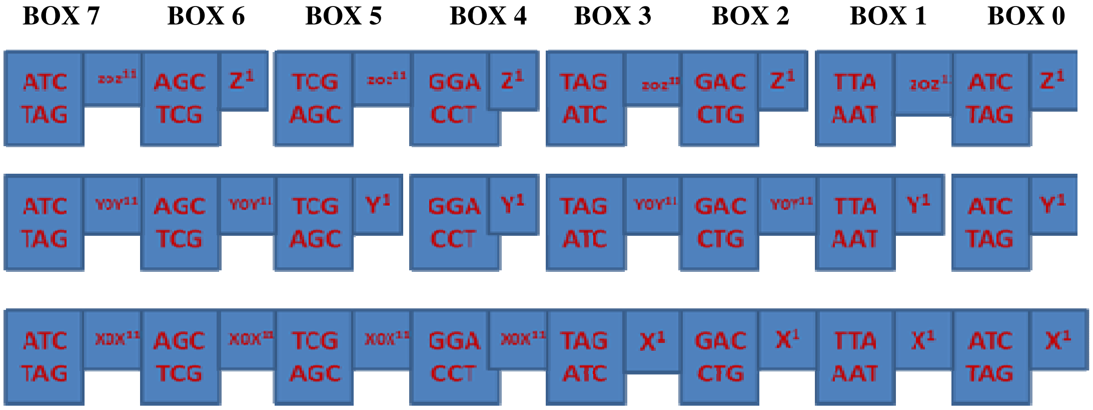
Step 3 and step 4 is repeated for second constraint equation , by adding the complementary strands $x1^{11}$,$x2^{11}$,$x3^{11}$ corresponding to variable x1,x2,x3. Any solution satisfying this inequality is hybridized with at most 1 complementary strand tagged with a fluorescent material. The feasible solution of the problem is " 4,2,1,0".


BOX 7
BOX 6
BOX 5
BOX 4
BOX 3
BOX 2
BOX 1
BOX 0
ATC TAG zoz11 AGC TCG Z1 TCG AGC zoz11 GGA CCT Z1 TAG ATC zoz11 GAC CTG Z1 TTA AAT zoz11 ATC TAG Z1
ATC TAG yoy11 AGC TCG yoy11 TCG AGC Y1 GGA CCT Y1 TAG ATC yoy11 GAC CTG yoy11 TTA AAT Y1 ATC TAG Y1
ATC TAG xox11 AGC TCG xox11 TCG AGC xox11 GGA CCT xox1 TAG ATC X1 GAC CTG X1 TTA AAT X1 ATC TAG X1

Hybridized of the Second constraint equation

STEP 6:

Step 3 and step 4 is repeated for third constraint equation , by adding the complementary strands $x1^{11}$,$x2^{11}$,$x3^{11}$ corresponding to variable x1,x2,x3.Any solution satisfying this inequality is hybridized with at most 3 complementary strand tagged with a fluorescent material. The feasible solution of the problem is "7,6,5,4,3,2,1,0".


BOX 7
BOX 6
BOX 5
BOX 4
BOX 3
BOX 2
BOX 1
BOX 0
ATC TAG zoz11 AGC TCG Z1 TCG AGC zoz11 GGA CCT Z1 TAG ATC zoz11 GAC CTG Z1 TTA AAT zoz1 ATC TAG Z1
ATC TAG yoy11 AGC TCG yoy11 TCG AGC Y1 GGA CCT Y1 TAG ATC yoy11 GAC CTG yoy11 TTA AAT Y1 ATC TAG Y1
ATC TAG xox11 AGC TCG xox11 TCG AGC xox11 GGA CCT xox11 TAG ATC X1 GAC CTG X1 TTA AAT X1 ATC TAG X1

Hybridized of the Third constraint equation

In above 3 constraints, we get four feasible solutions " 0,1,2,4". Then by comparing to the value of object function corresponding to every feasible solution,

	x^1	Y^1	Z
Box 1 –	0	0	1
	x^1	y	z^1
Box 2 –	0	1	0
	x	y^1	z^1
Box 4 –	1	0	0

Apply Box1 to objective function .

x^1 y^1 z

Max U = 4 *0 + 5*0 +4*1 → 0 0 1

= 0 + 0 + 4 = 4

Apply Box2 to objective function .

x^1 y z^1

Max U = 4 *0 + 5*1 +4*0 → 0 1 0

= 0 + 5 + 0 = 5

Apply Box4 to objective function .

x y^1 z^1

Max U = 4 *1 + 5*0 +4*0 → 1 0 0

= 4 + 0 + 0 = 4

This objective function is Maximized function. By comparing to the value of objective function corresponding to every feasible solution, we can obtain optimum solution (0,1,0) because its having maximum value of objective function is 5.

VI. Conclusion

This paper deals with a Bio-process to solve an Binary Linear programming problem using DNA computing approach. It introduces a solution based methods and the technique in general and the associate frame work accommodates a number of different feasible solutions to get the optimal. It also helps to make a best possible use of available productive resources such as time, labour, machine etc. In a production process, if bottlenecks occur the Linear Programming highlights the varieties of bottlenecks. Hybridization could be performed to extract the required computing output from the combination of Olingonucleotides. Since the applicability and feasibility of DNA computing approach, it is found that the more complex problems of this type of nature could be successfully designed. This problem can also be solved manually by using LPP Simplex method which also produces the same result.

References

[1] L.M.Adleman, Molecular Computations of solutions to combinatorial problems. Science, vol.266:pp.1021-1024,1994.

[2] L.M.Adleman, On constructing a molecular computer ftp://usc.edu/pub/csinfo/papers/adleman/molecular-computer-ps,University of Southern California, January 1995.

[3] L.M.Adleman, Computing with DNA Scientific American pp.34-41, 1998.

[4] E.Anne, W.Cai, M.Robert, The power of surface based DNA computation, Proceedings of the First Annual International conference on computational Molecular Biology, 1996.

[5] E.Baum and D.Boneh, Running Dynamic programming algorithms on a DNA computer,in a DNA based computers II (Landweber.F and Baum,E.B.,eds.), DIMACS series in Discrete mathematics and Theoretical Computer Science, vol.44, American Mathematical Society,Providence,RI, pp.87-96,1996.

[6] D.Boneh, C.Dunworth, J.Sgall. On the computational power of DNA. Discrete Applied Mathematics, vol.71(1-3): pp.79-94, 1996.

[7] D.Boneh, C.Dunworth, and R.J.Lipton. Breaking DFS using a molecular computer, in DNA based computers: Proceedings of a DIMACS workshop (Lipton.R.J and Baum.E.B.,eds.), American Mathematical Society, Providence, RI, pp.87-96,1996.

[8] D.Boneh, R.J.Lipton. Making DNA computers error resistant ftp://cs.princeton.edu/pub/people/dabo/bioerror.ps.z, 1996.

[9] D.Beaver, Computing with DNA.J.Comp.Biology, pp.1-7.

[10] R.Deaton,D. R.Francschetti, M.Garzon, J.A.Rose, R.C.Murphy and S.E.Stevens, " Information Transfer through hybridization reactions in DNA based computing " 1997.

[11] Eliezer L. Lozinskii, Boolean Modeling and computation, Journal on satisfiability 2006.

[12] M.Garzon, E.Eberbach, Dynamical Implementation of Nondeterministic Automata and concurrent systems.

[13] Gass.S.L, Linear Programming Methods and Applications, Newyork,1984.

[14] C.Gibas, P.Jambeck, Developing Bio informatics computer skills, o'Railly publications,2001.

[15] Gi - Joon Nam, K.A.Sakallah, R.A.Rutenbar, A Comparative study of two Boolean formulations of FPGA detailed routing constraints. June 2002.

[16] F.Guarinieri, M.Fliss, C. Bancroft, Making DNA add science, Vol. 273(5272): pp.220-223, July 1996.

[17] J.E. Hopcroft, J.F.Ullman , Introduction to Automata theory languages and computations. Addison Wesley,1979.

[18] J.E. Hopcroft, R.Motwani, J.F.Ullman , Introduction to Automata theory languages and computations. Addison Wesley,2000.

[19] E.Horrowitz, S.Sahani Fundamentals of computer Algorithms, Galgotia publications,1991.

[20] L.Kari , DNA computing : Arrival of Biological Mathematics, The Mathematical Intelligence, pp : 19,2,9-22,1997.

[21] L.Kari from Microsoft to Biosoft : Computing with DNA http:// www.csd.uwo.ca/~/ila,1997.

[22] R.J.Lipton, Using DNA to solve NP-complete problem, Technical Report, Princeton University 1995.

[23] R.J.Lipton, Using DNA to solve SAT . http//www.cs.princeton.edu/vdabo/bio- comp/sat.ps,1995.

[24] R.J.Lipton, DNA solution of hard computational problems science, Vol.268:pp.542-545, April 1995.

[25] R.J.Lipton .Speeding up computations via molecular biology, in DNA based computers (Lipton.R.J and Baum.E.B, eds) American Mathematical Society, providence, RI, pp.67-74,1996.

[26] Q.Liu, Z.Guo., A.Condon, R.Com M.Lagally and L.Smith, A surface – based approach to DNA computation, in DNa based computers II (L.F.Landweber and E.B.Baum, eds), DIMACS series in Discrete Mathematics and Theoretical computer science vol.44, American Mathematical Society, providence, RI, pp.123-132,1999.

[27] L.Streyer, Bio chemistry Freeman &co, 1995.

[28] Z.Tiina, formal models of DN computing a survey gaia.gi.eeleap/f49-2.3 pdf.

[29] Yoshida,H.Suyama, A DIMACS : series in Discrete Mathematics And theoretical computer science, solution to 3-SAT by Breadth-First search American Mathematical society, providence,2000.

[30] H.Y.Wu., An Improved surface based method for DNA computation, Biosystems, pp.59,1-5.